

# Navigation Based on Symbolic Space Models

Karolina Baras\*, A. Moreira\*\* and F. Meneses\*\*

\* Exact Sciences and Engineering Competence Centre, University of Madeira, Funchal, Portugal. Email: kbaras@uma.pt

\*\* Algoritmi Research Centre, University of Minho, Guimarães, Portugal. Email: {adriano, meneses}@dsi.uminho.pt

**Abstract**—Existing navigation systems are very appropriate for car navigation, but lack support for convenient pedestrian navigation and cannot be used indoors due to GPS limitations. In addition, the creation and the maintenance of the required models are costly and time consuming, and are usually based on proprietary data structures. In this paper we describe a navigation system based on a human inspired symbolic space model. We argue that symbolic space models are much easier to create and to maintain, and that they can support routing applications based on self-locating through the recognition of nearby features. Our symbolic space model is supported by a federation of servers where the spatial descriptions are stored, and which provide interfaces for feeding and querying the model. Local models residing in different servers may be connected between them, thus contributing to the system scalability.

**Keywords**—Symbolic Space Model; Indoor Navigation;

## I. INTRODUCTION

In recent years, there has been an effort to extend the convenience of outdoor navigation systems into the indoor environment. However, resorting to similar space models and routing algorithms has proved to be difficult and unpractical due to GPS limitations and the lack of geometric models for buildings.

There are, however, other systems where very simple models are used to assist navigation. One good example, where a simple topological model is used without any technological positioning system, is the underground transportation map. It is a graph in which vertices represent the stations and edges represent connections between them. In this system, positioning is based on the station names that are advertised at their entrances and inside the stations in places visible from inside the trains. Each user identifies his/her own location (graph node) according to the station name and defines a route by visual inspection of the lines (graph edges) connecting the stations.

In this paper we describe a navigation system that exploits human mental models of spaces [1] and self-location through the recognition of nearby features.

When we want to help someone to find a place inside a building we give them hints based on the current location. Usually we provide information like “walk through this corridor, go up two floors, turn left,...”. This kind of information is, in many cases, enough to guide people inside a building. A navigation application based on this

human behaviour has the potential of being more human friendly than other “more technological” approaches. Its development requires two major building blocks: a space model that mimics the human mental models, and a routing application. The remainder of this paper proposes a software solution for the implementation of these two components.

## II. SYMBOLIC SPACE MODEL

### A. Fundamentals

Similarly to human mental models of space, the model hereby proposed represents the environment as a set of interrelated objects. In this way it makes it easier for a human user to create his own model of the spaces he is familiar with, as an image of his mental model. As shown in Fig. 1, our model is a graph-based symbolic model in which vertices are objects and edges are relations among them. Each object may be related to several other objects that may be a part of the same local model or of a different model, as it will be explained further on.

In this model, each object is described by a set of attributes. Each relation is also characterized by a set of relation attributes. Object attributes and relation attributes are introduced to describe further the objects and the relations, and to store the time series of their dynamic characteristics. To support such dynamic nature of the model, each attribute is internally associated a timestamp.

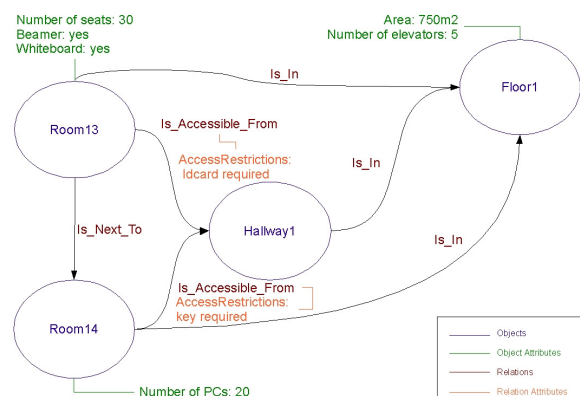


Figure 1. A snapshot of the symbolic world model

In navigation applications, attributes may help users to recognize space features, the same way commuters recognize their presence at a given underground station through the labels with the stations’ names. In addition, some of the attributes, both from objects and relations, may reflect time varying features of the space. An attribute named “open” and with values {yes, no} updated

This work was supported in part by Fundação para a Ciência e a Tecnologia under the grant SFRH/BD/36046/2007.

automatically may, for example, be used to reflect if a certain building entrance may be used or not depending on the time of the day. Similarly, an attribute of a relation, also updated automatically, may be used to reflect if a certain accessibility relation between two objects is active or not.

Basic definitions for the components of the proposed symbolic model are provided next.

**Definition 1:** An object is either a physical place (e.g., a room, a city square), a physical object (e.g., a table, an access point, a vehicle), an event (e.g., a conference) or an entity (e.g., an institution, a group of people). Each object is represented by a name, a type, a URI (Uniform Resource Identifier), consisting of an id, a host and a port and a set of attributes.

**Definition 2:** Object attributes are constant (e.g. the colour of an object or the area of a room) or time varying characteristics (e.g. the temperature) of an object. They are expressed as name-value pairs or name-value-timestamp triplets.

**Definition 3:** A relation is established between two objects and it represents their relationship in the real world.

**Definition 4:** A relation attribute exists to characterize a relation. It is a constant or time varying characteristic of a relation between two objects. It is expressed as a name-value pair or name-value-timestamp triplet.

#### B. Symbolic Contextualizer – the support for the model

The space model described above is supported by a federation of servers – Symbolic Contextualizers (SCs) – that store the objects and the relations between them, and provide an interface for feeding and querying the model. Fig.2 shows our system architecture which encompasses sensor networks, data acquisition and processing modules, world model services and applications.

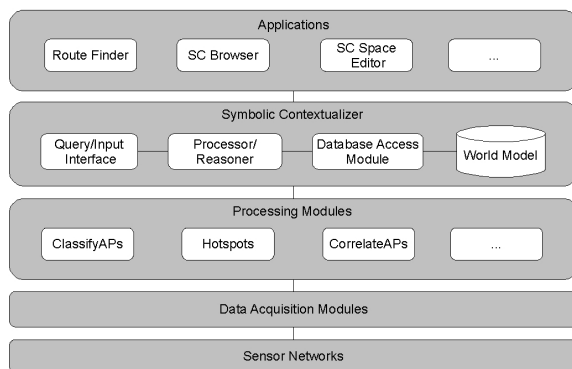


Figure 2. System architecture

The Symbolic Contextualizer input interface consists of a set of functions that allow for creating new objects and their descriptions, and corresponding relations in the model. Besides the name, type and URI, other internal data, such as an id, the date of creation and author, are stored by the Symbolic Contextualizer for operational reasons. The authorship of an object is also important to support the trust model associated with this approach (not described here).

Relations between objects in the model are similar to those created in people's minds, where links are established between the real world places as people visit them. The system has four basic predefined types of relations: *Is\_In*, *Is\_Accessible\_From*, *Is\_Next\_to* and *Is\_Near*. The *Is\_In* relation is transitive: if an object A is declared as being in B and B in C, then A is also contained in C. *Is\_Near* and *Is\_Next\_To* are always symmetric: if a relation is defined between A and B then it also exists implicitly between B and A. *Is\_Accessible\_From* can be either symmetric (bidirectional) or simple (unidirectional) allowing for the distinction of one way paths like the emergency exits or special areas in airports (e.g., passport control corridors). Based on these types of relations, inference algorithms were developed and may be used when querying the model to obtain a description of a place.

Besides these relations, users can create any other kind of relations, by defining for each one its name and type (transitive, symmetric or other). For example, a user can create an object that represents his house and create a relation *Is\_Visible\_From* to link the object to a viewpoint in his town.

The Symbolic Contextualizer query interface allows for querying the model to find the objects by name, by type, by id, by author, by relation name and by the value of an attribute. The query results depend on the inference level specified in the query. If, for example, transitivity is selected, then not only the selected objects are returned, but also all other objects that are related to them through a transitive type of relation. This behaviour makes it easier for applications to exploit the graph representation of the space model.

The dynamic nature of the real world is supported by the Symbolic Contextualizer in three different ways. The first one is by using external processing modules that collect data from sensors in the real world and automatically update attribute values, relations, or relation attribute values [2]. The second one is by using attributes with references to external data sources. In the current SC implementation this functionality is supported by the automatic processing of URLs (Uniform Resource Locator): whenever the value of an attribute is a URL, while processing a query the SC automatically replaces the URL by the value it returns. Here we assume that the URL contains the address of an external source which returns a meaningful value for an attribute. Therefore, connecting the space model to sensor networks is as easy as creating an attribute pointing to a web service. The third mechanism is by keeping track of the changes that occur in the values of attributes. Each update in the value of an attribute is stored as a new record with the corresponding timestamp. Storing the history of a place enables external applications to perform time series analysis for space characterization or anomaly detection, as described in our previous work [3].

### III. INDOOR NAVIGATION

The above described symbolic space model is part of a platform that supports context-aware applications. Its advantages for indoor navigation are manifold. First, it is much easier to understand by users and programmers because it is based on the human mental models. Second, each person can build his/her own model by creating objects and relations between them, without the need for

any geometric reference. Third, each local model can be stored on a local Symbolic Contextualizer, and objects in one server can be related with objects in other servers through their URIs (id, host, port triplets), thus contributing to the system scalability. Fourth, it supports routing applications that rely on a self-locating approach based on the recognition of visual landmarks (such as wall colours or door codes) described in the model as object attributes.

The whole navigation system includes the federation of Symbolic Contextualizers, a Space Editor for creating local models, a Browser for searching the models, and a routing application which will be briefly described in the following sections.

#### A. SC Space Editor

This is an editing wizard that allows for the creation of a new local space model as well as the enrichment of an already existing model. The user is identified by his user name and he has full permissions over the objects he creates. For a new object to be created, a name and a type are required. Names given to objects should be human readable and should correspond to the common name given to a place in order to make the model browsing easier and the self-location possible. Attributes and relations may be added to better describe the objects.

For creating a local model, users may choose to use an existing Symbolic Contextualizer, or install and maintain a personal or corporate SC, e.g., for privacy reasons. In any case, the local model can be linked to other public SCs to benefit from their models. This approach creates an opportunity for the collaborative creation of a global model, stored in a federation of distributed SCs.

The SC Space Editor has been implemented as a web application. A view of its user interface is shown in Fig.3.

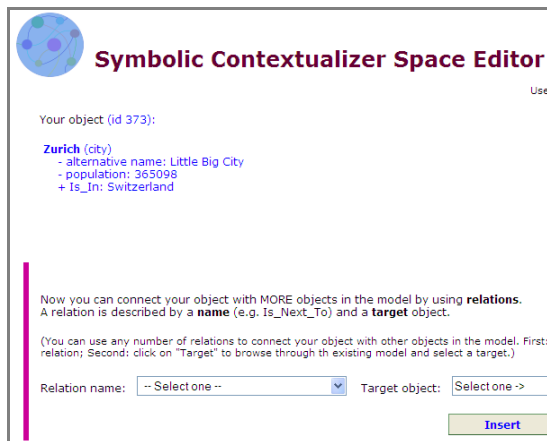


Figure 3. The web interface of the SC Space Editor

This application provides an interface where users create objects, their attributes and relations. Step by step, the user receives suggestions for types of objects, attribute names and relation names, based on the already existing elements in the model.

#### B. SC Browser

This application allows for browsing a symbolic space model starting with a name of one of the existing objects and following the links to other related objects. Instead of a map, the objects are presented in a list. Their

descriptions and the related objects are obtained by clicking on the links. In the graphic version of the application, the model is shown as a graph in which the objects are the vertices and the relations between objects are the edges. Further descriptions of the objects and relations are obtained by clicking on them.

#### C. The Route Finder Application

In order to help people to find a route from one place to another, we developed a web-based Route Finder application.

The application interface allows to specify the user current location and to define the desired destination. As for the SC Space Editor and SC Browser, it uses the Symbolic Contextualizer query interface to retrieve the descriptions of the selected objects. Based on the existing *Is\_Accessible\_From* relations, it calculates the shortest path between the two places.

The application provides the path the user has to follow if he wants to go from his current location to the specified destination. The path is the ordered list of places the user has to walk through, with the available description for each place (set of attributes). This solution allows the user to guide himself inside the building based on what he sees around. It is the digital version of the hints received when someone asks for directions. Fig.4 shows an example of the user interactions.

Initially, the user selects an origin and a destination, eventually using partial descriptions of the two places (Fig.4a). A place can be identified by different names which, in some cases, can cause some difficulties. In order to make this task easier, the interface provides a list of objects that match the users search. In this way, the user may select correctly his current location and destination (Fig.4b).

The Route Finder queries the Symbolic Contextualizer to find the shortest path between two places based on the *Is\_Accessible\_From* relation and its symmetry property. The shortest path is the one that leads the users to cross the smallest number of places.

The implemented algorithm that calculates the path was kept very simple at the cost of being less efficient. There are several options that may be used instead and that may be integrated in our solution in the future.

The algorithm starts with a search for the requested object descriptions in the model. Once found, it extracts all the relations *Is\_Accessible\_From* in which the destination object appears. The resulting list of existing links between pairs of related objects is saved and is used for further processing. The algorithm is recursive and it stops when the origin is found. When this happens, the final path from the origin to the destination is obtained from the list of linked objects and it is shown to the user (Fig.4c).

## IV. RELATED WORK

Significant research has been done concerning indoor positioning. Several authors suggest systems that use the WiFi network, RFID tags, ultrasound, infrared light and ultra wideband signals. Harle and Hopper [4] use an ultrasonic positioning system to evaluate their methods for discovering objects in the real world environment and building a world model out of it. Ruppel et al. [5] propose a way of modeling the topology of a building and a way to

combine WiFi indoor positioning system data with it. LaMarca et al. [6] present PlaceLab which enables devices like notebooks, PDAs and cell phones to locate themselves based on the WiFi, GSM and Bluetooth signals already available in the environment. Common to most of these systems is a geographic or 3D geometric model of the space.

a)

b)

c)

Figure 4. The interface of the RouteFinder application

In the research area of ubiquitous computing, location models have been developed mainly for tracking people and identifying the most suitable devices for the interaction with different context-aware applications. Leonhardt [7], Domnitcheva [8] and Becker and Dürr [9] systematize and describe different categories of location models (geometric, symbolic and hybrid). Examples of symbolic location models are given in [10, 11, 12].

In [9] the authors refer to graph-based symbolic world models as being adequate for nearest neighbour queries and navigation and lacking support for the range queries. In our model, the introduction of the semantic relations

between objects resolves this problem as we can define any kind of relation that is needed and associate to it a set of attributes to characterize it better (e.g. a distance measure between two places).

Stahl and Hauptert [13] describe these and some other issues related with indoor navigation, and propose a solution based on hybrid geometric/symbolic space models and indoor positioning systems. However, as in other approaches, this solution requires the building of geometric models for the buildings, which requires some level of technical expertise, access to CAD drawings, and a considerable amount of time (“... *the modelling of the rooms in the ground floor took a few hours.*” [13]). Resorting to the proposed concept of Symbolic Space Model and to the above described Space Editor, the effort to create the model for a building is considerably reduced.

## V. CONCLUSIONS AND FUTURE WORK

We briefly described an indoor routing platform supported by symbolic space models that can be created collaboratively by a group of users based on their mental space models. This approach represents an alternative to existing systems for indoor and pedestrian navigation that are based on geometric space model. One of the most appealing characteristics of this solution is that creating space models is much faster and do not require professionals with specialized technical skills.

Our future plans, to further enhance the routing capabilities, include the exploitation of other relations between objects, such as *Is\_Next\_To*, to improve the route description provided to the users. For computing estimates of the trip time, the values of some attributes, such as the physical length of objects or travel time attributes of *Is\_Accessible\_From* relations may also be explored.

Although the described approach is entirely supported by symbolic space models and human self-location, some navigation applications may benefit from automatic positioning to help users locate themselves. Integrating positioning systems providing position in a geometric referential into the symbolic space models framework is an open issue. One interesting approach for such automatic positioning is based on the recognition of radio signatures produced by WiFi and cellular (GSM) networks [14].

Another relevant functionality for pedestrian navigation support is the computation of travel times. This is certainly a complex task, far more difficult than the corresponding functionality available for car navigation. Again, try to mimic the human behaviour might be one interesting possibility.

The described applications may be found at the following links:

SC Browser: <http://local.dsi.uminho.pt/scb>

SC Editor: <http://local.dsi.uminho.pt/scse>

RouteFinder: <http://local.dsi.uminho.pt/route>

## REFERENCES

- [1] B.Tversky, Remembering spaces, in Endel Tulvig and Fergus I. M. Craik editors *The Oxford handbook of memory*, pp. 363-378, New York, USA, Oxford University Press, 2000.
- [2] K.Baras and A. Moreira, “Symbolic space modeling based on WiFi network data analysis”, in *2010 7th International Conference on Networked Sensing Systems*, June 2010.

- [3] K.Baras and A. Moreira, "Anomaly detection in university campus WiFi zones", in *2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pp. 202 -207, March 2010.
- [4] R.K.Harle and A.Hopper, "Dynamic world models from ray-tracing", in *Proceedings of the Second IEEE International Conference on Pervasive Computing and Communications (PerCom'04)*, pp. 55-64, 2004
- [5] P.Ruppel, C.Klein, C.Linnhoff-Popien, "Indooria – a platform for proactive indoor Location-based services", in *IEEE Globecom 2008 - Design & Developers Forum*, New Orleans, LA, USA, 2008.
- [6] A.Lamarca, Y.Chawathe, S.Consolvo, J. Hightower, I.Smith, J.Scott, T.Sohn, J.Howard, J.Hughes, F.Potter, J.Tabert, P.Powledge, G.Boriello, B.Schilit, "Place Lab: device positioning using radio beacons in the wild", in *Proceedings of the Third International Conference on Pervasive Computing*, pp. 106-133, Springer, 2005.
- [7] U.Leonhardt, *Supporting location-awareness in open distributed systems*, PhD Thesis, Department of Computing, Imperial College London, 1998.
- [8] S.Domnitcheva, "Location modeling: State of the art and challenges," in *Proceedings of the Workshop on Location Modeling for Ubiquitous Computing*, pp. 13–19, 2001.
- [9] C.Becker and F.Dürr, "On location models for ubiquitous computing," *Personal and Ubiquitous Computing*, vol. 9, pp 20-31, 2001.
- [10] B.Brumitt and S.Shafer, "Topological world modeling using semantic spaces", in *Proceedings of the Workshop on Location Modeling for Ubiquitous Computing*, pp. 55-61, 2001.
- [11] P.H.Hsieh and S.T.Yuan, "Dynamic semantic location modeling in mobile enterprise applications," in *ICEC '03:Proceedings of the 5<sup>th</sup> international conference on electronic commerce*, pp. 102-110 , New York, USA, 2003.
- [12] I.Satoh, "A spatial communication model for ubiquitous computing services", *Journal of Networks*, vol.3, issue 4, pp. 10-20, 2008.
- [13] C.Stahl and J.Hauptert, "Taking location modelling to new levels: A map modeling toolkit for intelligent environments," *2nd International Workshop on Location-and-Context-Awareness, LoCA 2006*, LNCS, 3987, pp. 74–85, 2006.
- [14] F.Meneses and A.Moreira, "Using GSM CellID Positioning for Place Discovering", in *Proceedings of First Workshop on Location Based Services for Health Care (LOCARE'06) at the 1st International Conference on Pervasive Computing Technologies for Healthcare 2006*, Innsbruck, Austria: November, 2006